Optical Flow and Surface Interpolation in Resistive Networks:

Algorithms and Analog VLSI Chips

Christof Koch[1,3], Jin Luo[1], James Hutchinson[2] and Carver Mead[1]

July 4, 1989

[1]Computation and Neural Systems Program, California Institute of Technology, Pasadena California, USA. [2] Thinking Machines Corporation, Cambridge, Massachusetts, USA. [3] To whom all correspondence should be addressed at Division of Biology, 216-76, Caltech, Pasadena, CA 91125, USA.

Abstract: To us, and to other biological organisms, vision seems effortless. We open our eyes and we "see" the world in all its color, brightness, and movement. Flies, frogs, cats, and humans can all equally well perceive a rapidly changing environment and act on it. Yet, we have great difficulties when trying to endow our machines with similar abilities. In this article, we describe recent developments in the theory of early vision that led from the formulation of the motion problem as an ill-posed one to its solution by minimizing certain "cost" functions. These cost or energy functions can be mapped onto simple analog and digital resistive networks. For instance, as detailed in this chapter, we can compute the optical flow by injecting currents into resistive networks and recording the resulting stationary voltage distribution at each node. These networks, which are implemented in subthreshold, analog, complementary metal oxide semiconductor (CMOS) very large scale integrated (VLSI) circuits, are very attractive for their technological potential.

1 Motion

The movement of objects relative to eyes or cameras serves as an important source of information for many tasks. We need motion to track objects and to determine whether an object is approaching or receding. Relative motion contains information regarding the three-dimensional structure of objects and allows biological organisms to navigate quickly and efficiently through the environment.

There exist two basic methods for computing motion (for reviews see Ullman, 1981; Hildreth, 1984; Hildreth and Koch, 1987). Intensity-based schemes rely on spatial and temporal gradients of the image intensity to compute the speed and the direction in which each point in the image moves. The output is a velocity or motion vector field covering the entire image. The second method is based on the identification of special features in the image, called *tokens*, which are then matched from image to image. This method relies on the unambiguous identification of the tokens—for instance, corners—in each image frame before the matching occurs and only yields a velocity vector at the sparse token locations. Psychophysical evidence suggests that both systems coexist in humans.

The principal drawback of all intensity-based schemes lies in the data used—temporal variations

in brightness patterns—which give rise to the perceived motion field, the *optical flow*. In general, the optical flow and the underlying velocity field, a purely geometrical concept, differ (Horn and Schunck, 1981; Verri and Poggio, 1989). For example, a featureless rotating sphere will not give rise to any optical flow, because the brightness does not appear to change even though the velocity field is non-zero. Conversely, if a shadow moves across the same featureless but now stationary sphere, the optical flow is non-zero although the velocity field is zero. Apart from such situations, the estimated optical flow will be close to the underlying velocity field, if strong enough gradients exist in the image. In this article, we assume that such strong gradients exist, as they do for most natural scenes, and consider how we can compute the velocity field using simple resistive networks.

1.1 Aperture problem

Let us derive an equation relating the change in image brightness to the motion of the image. We denote the image at time t by $I(x, y, t)$. Following Horn and Schunck (1981) we assume that the brightness of the image is constant over time:

$$\frac{dI}{dt} = 0. \tag{1}$$

This will be true, for instance, if a rigid object translates in space, but not if it rotates. On the basis of the chain rule of differentiation, eq. (1) transforms into

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = I_x u + I_y v + I_t = \nabla I \cdot v + I_t = 0, \tag{2}$$

where we define the velocity v as $(u, v) = (dx/dt, dy/dt)$, and where I_x, I_y and I_t are the partial derivatives of the brightness I with respect to x, y, and t. Because we assume that we can compute these spatial and temporal image gradients, we now have a single linear equation in two unknowns, u and v, the two components of the velocity vector. In other words, this equation by itself is not sufficient to uniquely determine the velocity field. Figure 1 graphically illustrates this *aperture problem*. Any measuring system with a finite aperture obeying the brightness constrain equation (1), whether biological or artificial, can only measure the velocity component perpendicular to the edge or along the spatial gradient ($-I_t / \mid \nabla I \mid$). The

component of motion perpendicular to the gradient cannot, in principle, be registered. The problem remains unchanged even if we measure these velocity components at many points throughout the image. For each measurement, we recover one equation with two unknowns. If a rigid object is translating through space, two local motion estimates (in general) are sufficient to uniquely determine the true velocity (Hildreth, 1984). For general motion of non-rigid objects, such as people, however, where the velocity field can vary from location to location, additional constraints must be introduced. [1]

1.2 Smoothness Assumption

Formally, this problem can be characterized as ill-posed (Poggio, Torre and Koch, 1985). Hadamard introduced this concept to describe problems in mathematical physics that either (1) have no solution at all, (2) have no unique solution, or (3) do not depend continuously on the initial data. Inverse problems, such as computer-tomography, are instances of ill-posed problems. All problems in *early vision*, which we define as the set of processes that recover the properties of the visible three-dimensional surfaces from the two- dimensional intensity arrays on retinae or cameras, are ill-posed. For example, binocular stereo and interpolating surfaces from sparse and noisy data are ill-posed, because in the former many and in the latter infinitely many solutions exist.

How can we make these problems well-posed, with unique solutions depending continuously on the data? One method of "regularizing" ill-posed problems involves restricting the class of admissible solutions by imposing appropriate constraints (Poggio *et al.*, 1985). Applying this method to motion, one can argue that, in general, objects are smooth—except at isolated discontinuities—undergoing smooth movements. Thus, in general, neighboring points in the world will have similar velocities. The projected velocity field should reflect this fact. We therefore impose on the velocity field the constraint that it should be the smoothest (in a specified technical sense) while satisfying the data. As the measure of smoothness we follow Horn and Schunck (1981) and choose the square

[1] Recently, Verri, Girosi and Torre (1989) exploited a different constraint equation to yield a unique value of the optical flow at every location. However, since this measure depends on computing second derivatives, a smoothing stage, as discussed below, still has to be used.

of the velocity field gradient:

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2. \tag{3}$$

A variational functional provides the most general way to formulate the problem (Horn and Schunck, 1981): the final velocity field (u, v) should minimize

$$E(u,v) = \iint (I_x u + I_y v + I_t)^2 + \lambda \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] dx\, dy, \tag{4}$$

where the regularization parameter λ is inversely dependent on the signal-to-noise ratio. The first term describes the fact that the final solution should follow as closely as possible the measured data whereas the second term imposes the smoothness constraint on the solution. The degree of minimization of one or the other

term is governed by λ. With accurate data, violating the first term should be "expensive" and λ will be small. Conversely, with unreliable data (low signal-to-noise ratio), much more emphasis will be placed on the smoothness term.

The "energy functional" $E(u, v)$ is quadratic in the unknown u and v. It then follows from standard calculus of variation that the associated Euler-Lagrange equations will be linear in u and v:

$$\begin{aligned} I_x^2 u + I_x I_y v - \lambda \nabla^2 u + I_x I_t &= 0 \\ I_x I_y u + I_y^2 v - \lambda \nabla^2 v + I_y I_t &= 0 \end{aligned} \tag{5}$$

We now have two linear equations at every point. Our problem is therefore completely determined. We could now use a number of iterative numerical techniques, such as steepest descent, to solve these equations for u and v. Instead, we pursue a different path.

1.3 Analog Resistive Networks

Let us assume that we are formulating eqs. (4) and (5) on a discrete two-dimensional grid, such as the one shown in Figure 2a. Equation 5 then transforms into

$$\begin{aligned} I_{xij}^2 u_{ij} + I_{xij} I_{yij} v_{ij} - \lambda (u_{i+1j} + u_{ij+1} - 4u_{ij} + u_{i-1j} + u_{ij-1}) + I_{xij} I_{tij} &= 0 \\ I_{xij} I_{yij} u_{ij} + I_{yij}^2 v_{ij} - \lambda (v_{i+1j} + v_{ij+1} - 4v_{ij} + v_{i-1j} + v_{ij-1}) + I_{yij} I_{tij} &= 0, \end{aligned} \tag{6}$$

where we replaced the Laplacian with its simplest four-point approximation on a rectangular grid. We now show that this set of linear equations can be solved naturally using a simple resistive network. Let us apply Kirchhoff's current law to the center node of the resistive network shown in Figure 2b. We then have the following update equation:

$$C\frac{du_{ij}}{dt} = T\left(u_{i+1j} + u_{ij+1} - 4u_{ij} + u_{i-1j} + u_{ij-1}\right) + g^u_{ij}\left(E_{ij} - u_{ij}\right), \tag{7}$$

where the u_{ij}'s represent voltages, E_{ij} corresponds to a battery, T and g_{ij} to conductances and C to a capacity. Let us now assume that we have two such resistive networks superimposed, with the node ij in the upmost network connected—via a conductance T_{c-ij}—to the appropriate node ij in the bottom network (see Figure 3b). We then have two equations similar to eq. (7), with a coupling term $T_{c-ij}(v_{ij} - u_{ij})$, where v_{ij} is the voltage at node ij in the bottom network. If we assume that the resistive network has converged to its final state, $du_{ij}/dt = 0$ and $dv_{ij}/dt = 0$, we see that both equations are identical with eq. (6), if we identify

$$\begin{aligned}
T &\longrightarrow \lambda \\
T_{c-ij} &\longrightarrow -I_{xij}I_{yij} \\
g^u_{ij} &\longrightarrow I_{xij}\left(I_{xij} + I_{yij}\right) \\
g^v_{ij} &\longrightarrow I_{yij}\left(I_{xij} + I_{yij}\right) \\
E_{ij} &\longrightarrow \frac{-I_t}{I_{xij} + I_{yij}}.
\end{aligned} \tag{8}$$

Once we set the batteries and the conductances to the values indicated in eq. (8), the network will settle into the state of least power dissipation. The key idea is that because the electrical power in a linear network is proportional to the square of the voltage gradient across all resistances, the electrical power dissipated by the circuit in Figure 3b can be identified with the variational functional of eq. (4). *Maxwell's Minimum Heat Theorem* (Maxwell, 1891) then implies that in such an idealized linear network the distribution of voltages and currents (subject to Kirchhoff's laws) minimizes the total power dissipated as heat.[2] The associated stationary voltages correspond to the

[2]Note that, in a *abuse de language*, the functional E of eq. (4) is frequently termed "energy", while it actually represents "electrical power" in our case.

solution sought: u_{ij} is equivalent to the x component and v_{ij} to the y component of the optical flow field. A unique and stable solution alway exists, even if some of the conductances have negative values. In fact, many of the conductances connecting the lower and the upper networks (T_{c-ij}) and the conductances associated with the batteries (g^u_{ij} and g^v_{ij}) will be negative, because the sign of I_x and I_y can be either negative or positive. As we will see, this poses no serious problems, given the technology we have chosen to build resistances.

We simulated the behavior of these networks for both synthetic and natural images by solving the previous circuit equations at each node. As boundary conditions, we copied the initial velocity data at the edge of the image into the nodes lying directly adjacent to but outside the image (zero normal derivative). We estimated the spatial and temporal derivatives I_x, I_y, and I_t using a discrete eight-point approximation.

Given the high computational cost associated with solving these elliptical equations, we use parallel computers of the Hypercube family, the 32-node Mark III Hypercube at the Jet Propulsion Laboratory and a 4- and a 16-node NCUBE in our laboratory at Caltech. Even though we use a variable time-step algorithm, convergence times are slow (10 minutes for a 128 x 128 image). Solving eq. (5) is similar to solving Poisson's equation; thus, the number of iterations required to converge is proportional to n^2 (on an n x n-pixel image).

The sequences in Figures 4, 5, and 6 illustrate the resulting optical flow. Two synthetic images (Fig. 4a,b) contain two squares moving diagonally past each other. Figure 4c illustrates the initial velocity data and the velocity component perpendicular to the image gradient. Figure 4d shows the resulting smooth optical flow. As discussed by Horn and Schunck (1981) the smoothness constraint leads to a qualitatively correct estimate of the velocity field. Thus, one undifferentiated blob appears to move to the lower right and one blob to the upper left. However, at the occluding edge where both squares overlap, the smoothness assumption results in a spatial average of the two opposing velocities, and the estimated velocity is very small or zero. In parts of the image where the brightness gradient is zero and thus no initial velocity data exist (for instance, in the interiors of the two squares), the velocity estimate is simply the spatial average of the neighboring velocity

estimates. These empty areas eventually will fill in from the boundary, similar to the flow of heat for a uniform flat plate with "hot" boundaries.

The image sequence in Figure 5 illustrates the effect of varying the conductance T between neighboring points. As we place more confidence in the measured data (small λ), the coupling between neighboring nodes decreases because T decreases, and the optical flow becomes more inhomogeneous, better reflecting the correct velocity field. As the data become less reliable (large λ), more smoothing occurs until little spatial variation exists.

1.4 Motion Discontinuities

The smoothness assumption of Horn and Schunck (1981) regularizes the aperture problem and leads to the qualitatively correct velocity field inside moving objects. However, this approach smoothes over figure-ground discontinuities or completely fails to detect the boundary between two objects with differing velocities, because the algorithm combines velocity information across motion boundaries. We argue that motion discontinuities are the most interesting locations in any image, because they indicate where one object ends and another one begins. Motion as well as intensity discontinuities are vital for solving the critical object segmentation problem that occurs at a subsequent stage of the image understanding process.

Various researchers have attempted to prevent the smoothing constraint from take effect across strong velocity gradients (e.g. Nagel, 1987). Geman and Geman (1984) proposed a successful strategy for dealing with discontinuities. They exploited an analogy between statistical mechanics and images, whereby the intensity values at each picture element and the presence or absence of discontinuities are viewed as states of particles on a lattice. We can assign an "energy" function to this system and compute its most likely state.

In this article, we do not rigorously develop this approach, based on Bayesian estimation theory (Geman and Geman, 1984; Marroquin, Mitter and Poggio, 1987). Suffice it to say that a priori knowledge (for instance, that the velocity field in general should be smooth) can be formulated in terms of a Markov random field model of the image (in a Markov random field, the conditional probability that a given variable at location i, j has a particular value f_{ij} depends only on the

values of f in a neighborhood of i, j). Given such an image model, and given noisy data, we then estimate the "best" flow field by some likelihood criterion. The one we use here is the *maximum a posteriori* estimate, although other possible criteria have certain advantages (Marroquin *et al.*, 1987). Maximizing the *a posteriori* probability yields the best solution, which is fully equivalent to minimizing an expression such as eq. (4).

To reconstruct images consisting of piecewise constant segments, Geman and Geman (1987) further introduced the powerful idea of a line process l. Their stochastic approach was extended and modified to account for discontinuities in depth, texture, optical flow and color (Marroquin *et al.*, 1987; Hutchinson, Koch, Luo and Mead, 1988; Poggio*et al.*, 1988; Nasrabadi, Clifford and Liu, 1989). For our purposes, we will assume that a line process can occupy one of two states: "on" ($l = 1$) or "off" ($l = 0$). Line discontinuities are located on a regular lattice set between the original pixel lattice (see Figure 3a), such that each pixel ij has one horizontal l^h_{ij} and one vertical l^v_{ij} line process associated with it. If the appropriate line process is turned on, the smoothness term between the two adjacent pixels will be set to zero.

To prevent line processes from forming everywhere and to incorporate additional knowledge regarding discontinuities into the line processes, we must include an additional term $V_c(l)$ in the new energy functional:

$$\begin{aligned} E(u, v, l^h, l^v) \;=\; & \sum_{i,j} (I_x u_{ij} + I_y v_{ij} + I_t)^2 \\ & + \lambda \sum_{i,j} \left(1 - l^h_{ij}\right) \left[(u_{i+1j} - u_{ij})^2 + (v_{i+1j} - v_{ij})^2\right] \\ & + \lambda \sum_{i,j} \left(1 - l^v_{ij}\right) \left[(u_{ij+1} - u_{ij})^2 + (v_{ij+1} - v_{ij})^2\right] \\ & + V_c(l). \end{aligned} \tag{9}$$

V_c contains a number of terms penalizing or encouraging specific configurations of line processes:

$$V_c(l) = C_c \sum_{i,j} l^h_{ij} + C_p \sum_{i,j} l^h_{ij} \left(l^h_{ij+1} + l^h_{ij+2}\right) + C_I V_I(l), \tag{10}$$

plus the corresponding expression for the vertical line process l^v_{ij} (obtained by interchanging i with j and l^v_{ij} with l^h_{ij}). The first term (C_c) penalizes each introduction of a line process, because the

cost C_c has to be "paid" every time a line process is turned on. The second term prevents the formation of parallel lines: if either l^h_{ij+1} or l^h_{ij+2} is turned on, this term will tend to prevent l^h_{ij} from turning on. The third term (C_I) embodies the fact that, in general, motion discontinuities occur along extended contours and rarely intersect. We adopt the function given by Koch *et al.* (1986) favoring the formation of motion discontinuities along extended contours and penalizing both multiple line intersections and isolated discontinuities.

We obtain the optical flow by minimizing the cost function in eq. (9) with respect to both the velocity field (u, v) and the line processes l^h and l^v. However, unlike before, this cost or energy functional is non-convex, since it contains cubic terms (e.g. $l^h_{ij}u^2_{ij}$). Geman and Geman (1984) resorted to annealing, a statistical optimization technique, to find the ground state of their system. If annealing is applied appropriately, the system converges with probability converging to one to the global maximum. However, the length of the required convergence times makes any practical application expensive. A number of authors have used deterministic methods to find the (local) minima of the associated non-convex variational functionals, with next-to-optimal results (e.g. Terzopoulos, 1986; Koch, Marroquin and Yuille, 1986). A rigorous deterministic approach has been championed by Blake and Zisserman (1987). Their "graduated non-convexity" (GNC) algorithm bears many similarities to the above methods, and leads to excellent results in the case of piece-wise continuous reconstruction of surfaces (Blake, 1989).

We use a deterministic approximation of the stochastic search algorithm based on solving Kirchhoff's equations for a mixed analog and digital network. Our algorithm exploits the fact that for a fixed distribution of line processes, the functional of eq. (9) is quadratic. Thus, in the analog cycle, we first initialize the resistive network (see Figure 3b) according to eq. (8) with all line processes switched off, i.e. $l_{ij} = 0$ for all i, j. The network then converges to the smoothest solution. Subsequently, in the digital cycle, we update the line processes by deciding at each site of the line process lattice whether the overall energy can be lowered by setting or breaking the line process. We always accept the state of the line process corresponding to the lower energy configuration: l^h_{ij} will be turned on if $E(u, v, l^h_{ij} = 1, l^v) < E(u, v, l^h_{ij} = 0, l^v)$; otherwise, $l^h_{ij} = 0$. This computation

only requires local information. Breaking the appropriate resistive connection between the two neighboring nodes switches on the line processes. After the completion of one such analog-digital cycle, we reiterate and compute the smoothest state of the analog network for the newly updated distribution of line processes.

Although we have no guarantee that the system will converge to the global minimum, given our use of a gradient descent rule in a non-convex system, this deterministic algorithm seems to find next-to-optimal solutions (see Figures 4, 5, and 6) in about 10 to 15 analog-digital cycles. Furthermore, the algorithm must converge, because at each step E is always reduced and E is bounded from below. We compared statistical annealing with our deterministic method in the case of interpolating and smoothing sparsely sampled data in the presence of discontinuity, where the underlying energy function is similar to E in eq. (9). Both methods converged to qualitatively similar solutions (Koch *et al.*, 1986).

The synthetic motion sequence in Figure 4 demonstrates the dramatic effect of the line processes. The optical flow outside the discontinuities approximately delineating the boundaries of the moving squares is zero, as it should be (see Figure 4e). Where the two squares overlap, however, the velocity gradient is high and multiple intersecting discontinuities exist.

To restrict further the location of discontinuities, we adopt a technique used by Gamble and Poggio (1987; see also Poggio *et al.*, 1988) to locate depth discontinuities by requiring that our motion discontinuities coincide with the location of intensity discontinuities or edges. In general, the physical processes and the geometry of the three-dimensional scene giving rise to the motion discontinuity will also give rise to an intensity edge. For example, moving physical objects occluding other objects will give rise to an image with edges at the occluding boundaries. In fact, only under laboratory conditions—for instance, using random dot patterns— does a motion discontinuity not coincide with intensity edges. As edges we use the zero-crossings of a Laplacian of a Gaussian convolved with the original image. Marr and Hildreth (1980) have shown that these locations usually correspond to physical edges. We threshold these zero-crossings (based on the square of the gradient) in order to remove spurious zero-crossings caused by noise and "weak" edges. Other

edge detection algorithms should work equally well.

We now add a new term $V_{Z-C_{i,j}}$ to our energy function E, such that $V_{Z-C_{i,j}}$ is zero if $l_{i,j}$ is off or if $l_{i,j}$ is on and a zero-crossing exists between locations i and j. If $l_{i,j} = 1$ in the absence of a zero-crossing, $V_{Z-C_{i,j}}$ is set to a large positive number (in our case 1000). This strategy effectively prevents motion discontinuities from forming at locations where no zero-crossings exist, unless strongly suggested by the data. Conversely, however, zero-crossings by themselves will not induce the formation of discontinuities in the absence of motion gradients.

Figure 4f demonstrates that this strategy leads to the correct velocity field—with the exception of the corners—in addition to labeling all motion discontinuities. Figures 5 and 6 demonstrate our method on image pairs obtained with a video camera. Our recent experience (Harris, Koch, Staats and Luo, 1989) has show that the co-localization of motion discontinuities with intensity edges makes most of the additional constraints we and other use redundant. Specifically, because zero-crossings are always continuous (except at boundaries) and never intersect, we do not require the line potential term $V_l(l)$ in eq. (10), nor the C_p term preventing parallel line formation. The C_c term then controlls if and how discontinuities are "created" .

1.4.1 Varying the 'amplitude' of motion discontinuities

When dealing with real data, the amplitude of velocity and, consequently, the amplitude of any motion discontinuity vary over a considerable range (as compared to the artificial situation in Figure 4). Because we assume no *a priori* knowledge concerning the distribution and amplitude of any motion discontinuities, we adopted the following heuristic procedure. Similar to Marroquin (1985), Terzopoulos (1986) and Koch *et al.* (1986) we vary the magnitude of the V_C term in eq. (9) by multiplying V_C with $1/K(t)$. Initially, $K(t)$ is small, but it then increases linearly until a given upper bound. In other words, the formation of discontinuities is penalized initially, encouraging a smooth interpolation everywhere except at very steep velocity gradients. Subsequently, by paying a smaller price for the formation of line processes, the optical flow will break at smaller velocity gradients. The final state of the network is independent of the speed at which $K(t)$ changes

(adiabatic convergence). All other parameters remain constant and are identical for all simulations reported in this article.

2 Analog VLSI Networks

Even with the approximations and optimizations we previously described, the computations involved in this and similar early vision tasks require seconds to many minutes on even large-scale parallel computers. For the computations to be truly useful, we should be able to carry them out on a whole image in real time. Fortunately, modern integrated circuit technology gives us a medium in which we can realize extremely complex, analog real-time implementations of these computational metaphors (Mead, 1989).

We can achieve a compact implementation of a resistive network using an ordinary CMOS process, provided the transistors run in the subthreshold range, where their characteristics are ideal for implementing low-current analog functions. We achieve the effect of a resistor by choosing the circuit configuration shown in Figure 7 rather than by using the resistance of a special layer in the process. We can control the value of the resulting resistance over five orders of magnitude by setting the bias voltages on the upper and lower current source transistors. The current-voltage curve saturates above approximately 100 mV, a feature that we can use to advantage in many applications.

With small voltage gradients, we can treat the circuit as if it were a linear resistor, as shown by the shaded areas on the curves (Figure 7b). Conductances to signal input sources are implemented with transconductance amplifier followers, as shown in Figure 7a. Each amplifier injects a current into the network proportional to the difference between the local signal potential and the potential of the network. The effect of a conductance is thus achieved without drawing any current from the signal source. The value of the conductance is set by the transconductance control on the amplifier, which we can use to reflect the confidence assigned to the particular input. High conductance values give the network a short spatial-averaging scale, low values give a long averaging scale. Figure 7c shows the spatial response of an experimental one-dimensional network to a point stimulus.

We obtained the different values of the space constant L by appropriate settings of the amplifier transconductances. We can easily realize resistances with an effective negative resistance value.

A 48 by 48 silicon retina has been constructed that uses the hexagonal network of Figure 8 as a model for the horizontal cell layer in the vertebrate retina (Sivilotti, Mahowald amd Mead, 1987; Mead, 1989). In this application, the input potentials were the outputs of logarithmic photoreceptors—implemented via phototransistors—and the potential difference across the conductance T formed an excellent approximation to the Laplacian operator (Marr and Hildreth, 1981). This model results in the classical center-surround receptive field properties observed in the response of retinal ganglion cells. The circuit performs in real time. Using the same computational elements, Tanner (1986; see also Mead, 1989) build a simple motion chip implementing the brightness constraint eq. (2) for rigid motion. This circuit with an array of 8 by 8 photoreceptors (acting similar to an optical "mouse") computes the constant (in space) optical flow induced by motion of a rigid body and essentially corresponds to the network in Figure 3b with all horizontal resistances set to zero. We will now discuss the performance of a circuit implementing the same type of smoothness operator underlying the optical flow computation.

2.1 Resistive Network for Smooth Surface Interpolation

The simplest variational functional for smooth surface interpolation is given by (Grimson, 1981):

$$E(f) = \lambda \sum_{ij} \left((f_{i+1j} - f_{ij})^2 + (f_{ij+1} - f_{ij})^2 \right) + \frac{1}{2\sigma^2} \sum_{ij} (f_{ij} - d_{ij})^2, \tag{11}$$

where the second sum only contains locations ij where data exists. The principal difference to the optical flow computation is the manner in which the final solution is constrained by the measured data ($(f - d)$ versus the brightness constraint equation), while the smoothness term is identical to the one used in eq. (4). The functional E can be considered to be the power dissipated in the resistive network shown in Figure 8 if the battery V_{in} is set to the measured depth value d_{ij} and the value of the conductance G is set to $1/(2\sigma^2)$, where σ^2 is the variance of the Gaussian noise process associated with the depth measurements. If no depth data is present at a particular node, G is set to zero at that location. The stationary state of this network— dictated by Kirchhoff's laws—then

corresponds to the global minimum of E in eq. (11) and the voltage at each node corresponds to the discrete values of the smoothed surface fitted through the noisy and sparse measurements.

Each point on the hexagonal grid in Figure 8 is coupled to six equivalent neighbors. The high degree of symmetry of such an arrangement creates a nearly isotropic environment, free of many of the "preferred axis artifacts" introduced by an orthogonal grid. In addition, the larger connectivity allows a greater variation in effective resistor value caused by variations in transistor parameters.

Figure 9 shows an analog CMOS VLSI test chip implementing this network on a 48 by 48 grid. Each node includes the resistor apparatus and a set of sample-and-hold circuits for setting the confidence and signal input voltages. In addition, an output amplifier enables measurement of the node voltage without disturbing the node itself. A scanning mechanism addresses both the sample-and-hold circuits and the output buffer, so the stored variables can be refreshed or updated, and the map of node voltages can be read out in real time.

Experimental data for this 100,000 transistor 48 by 48 pixel circuit, build via MOSIS, is illustrated in Figures 10, 11 and 12. The response of this two-dimensional network to a thin but extended line (a sort of one-dimensional Green function) is plotted in Figure 10, while the last two figures shown the smoothing and filling properties of this chip using the same data set, but for two different boundary conditions. In Figure 11 the voltage is set to zero along the boundary; in other words, the surface f is "pinned" to ground. In Figure 12, on the other hand, we impose a zero-slope condition, i.e. no current flows through the boundary nodes, effectively allowing the membrane to freely "float". We have not yet characterized in detail the accuracy (in terms of bits of resolution) of this network.

3 Discussion

We have demonstrated that the introduction of binary motion discontinuities into Horn and Schunck's (1981) algorithm leads to a much improved performance of their method, particularly for the optical flow in the presence of a number of moving objects. Moreover, we have shown that the appropriate computations map onto simple resistive networks.

We are now implementing these resistive networks into VLSI circuits, using subthreshold CMOS technology. We briefly illustrated the behavior of one such circuit, smoothing sparsely sampled and noisy depth data. We have also demonstrated a working chip implementing an analog version of binary line discontinuities (via a "resistive fuse"; see Harris, Koch, Luo and Wyatt, 1989). We thus have all the elementary circuit elements in hand—phototransistors for on-chip image acquisition (Mead, 1989), resistive networks for smoothing, and resistive fuses for detecting discontinuities—to build analog chips for a large number of early vision tasks, such as binocular stereo, optical flow, edge detection, surface interpolation, and structure from motion (for a review see Koch,1989). All these networks share several features with biological neuronal networks. Specifically, they do not require a system-wide clock, they rely on many connections between simple computational nodes, they converge rapidly (within several time constants), and they are quite robust to hardware errors. Furthermore, our networks consume moderate amounts of power, because each resistive element operates in the millivolt and 10 nanoampere range. The entire retina chip (Mead, 1989) requires about 100 μW (the dominant power consumption lies in the photoconversion stage).

These features—real-time performance, low power consumption, robustness, and small spatial dimensions—make these circuits attractive for a variety of deep space missions. In collaboration with the Jet Propulsion Laboratory, we are currently evaluating the feasibility of such resistive network-based vision systems for autonomous vehicles to be used in the exploration of planetary surfaces, such as that of Mars.

Acknowledgments An early version of this model was developed and implemented in collaboration with A.L. Yuille (1987). Matthew Avalos and Andrew Hsu wrote the code for the Imaging Technology system, and Eric Staats for the Ncube. C.K. is supported by an Office of Naval Research Young Investigator Award and grants from the NSF Advanced Engineering Program (EET-8714710 and IST-8700064), the James Lee Powell Foundation and Rockwell International Science Center. C.M. is supported by the Office of Naval Research and the System Development Foundation.

4 Figure Legends

Figure 1. The aperture problem of motion. Any system with finite aperture, whether of biological or artificial origin, obeying the image brightness constraint equation $dI/dt = 0$, can only measure the velocity component $-I_t/|\nabla I|$ along the spatial gradient ∇I. Motion perpendicular to the gradient will not be visible, except by tracking salient features in the image (Horn and Schunck, 1981; Hildreth, 1984).

Figure 2. Rectangular grid and resistive network for surface interpolation. (a) Rectangular grid for solving the discrete version of eq. (4). (b) Part of the resistive network minimizing the discrete approximation of the variational functional E in eq. (4). We assume the conductance T connecting neighboring nodes to be constant. Each node connects to a variable battery E_{ij} via a conductance g^u_{ij}. Parasitic capacities (C) give the circuit its dynamic behavior. The final network (see Figure 3b) consists of two such resistive networks superimposed, where corresponding nodes are connected via a variable conductance T_{c-ij}, as in Figure 3b. Once the batteries E_{ij} and conductances g^u_{ij} and g^v_{ij} have been set, the network will converge—following Kirchhoff's laws—to the state of least power dissipation that corresponds to the solution of the variational eq. (4).

Figure 3. Rectangular grid with line processes and hybrid network. (a) The location of the horizontal (l^h_{ij}) and vertical (l^v_{ij}) line processes relative to the rectangular motion-field grid. (b) The hybrid resistive network, computing the optical flow in the presence of binary discontinuities. The conductances T_{c-ij}, connecting both grids depend on the brightness gradient, as do the conductances g^u_{ij} and g^v_{ij}, which connect each node with the battery. For clarity, only two such elements are shown. The battery E_{ij} depends on both the temporal and the spatial gradient and is zero if no brightness change occurs. The x component of the velocity, u, is given by the voltage in the top network, while the y component of the velocity, v, is given by the voltage in the bottom network. A high voltage value at location i, j will spread to its four neighboring nodes. The degree

to which voltage spreads depends on the value of the fixed conductance, T, given by the inverse of the signal-to-noise ratio. Binary switches, which make or break the resistive connections between nodes, implement motion discontinuities, because an arbitrary high voltage (velocity) will not affect the neighboring site across the discontinuity. An extended horizontal motion discontinuity is indicated. We have build analog circuit implementations of these discontinuities (Harris *et al.*, 1989).

Figure 4. Motion sequence using synthetic data. (a) and (b) Two 32 by 32-pixel images of three high-contrast squares on a homogeneous white background. Only the two squares on the upper left are displaced. (c) The initial velocity data. The insides of both squares contain no data. (d) The final state of the network after 240 iterations, corresponding to the smooth optical flow field. The algorithm gives a qualitatively correct estimate of the velocity field. Note, however, the vanishing velocity estimate at the occluding edges where the two moving squares overlap, caused by the averaging property of the smoothness constraint. Moreover, the moving objects are not delineated in the flow field, because the algorithm smooths over the figure-ground motion discontinuity. (e) Optical flow in the presence of motion discontinuities (indicated by solid lines). Numerous line processes are turned on in the area where the moving objects overlap. The formation of discontinuities along continuous contours is explicitly encouraged. (f) Discontinuities are strongly encouraged to form at the location of intensity edges. This additional constraint leads to the correct velocity field. The location of these discontinuities facilitates object segmentation at a later stage of visual analysis. Both (e) and (f) show the state of the hybrid network after six analog-digital cycles.

Figure 5. Optical flow of a moving hand. (a) and (b) Two 128 x 128 pixel images captured by a video camera. The hand is displaced downward by up to two pixels. (c) Zero-crossings of the Laplacian of a Gaussian (with seven-pixel-wide center) superimposed on the initial velocity data. The zero-crossings are thresholded to remove noise. In areas with little or no spatial gradients,

amplified image noiose leads to noisy velocity data, since their amplitude is given by $-I_t/|\nabla I|$. The zero-crossings of both images are shown, thus the double line. Notice the stationary zero-crossing at the right edge. (d) The smooth optical flow after 1000 iterations. In these and the following images, the plotted individual velocity vectors are not highly visible; however, the gray-scale intensity is proportional to the magnitude of the velocity (the direction is always downward). The smooth optical flow for a five-times-lower and five-times-higher value of the conductance T appears in (e) and (f). The next four images show the state of the hybrid network after the first (g), second (h), fifth (i), and ninth (j) analog-digital cycles. In the final image, the fingers have a higher velocity than does the hand itself. It takes several cycles for the motion discontinuities to "creep" around the outline of the hand.

Figure 6. Optical flow of a moving person. (a) and (b) Two 128 x 128-pixel images captured by a video camera. The person in the foreground is moving toward the right while the person in the background is stationary. The noise in the lower part of the image is a camera artifact. (c) Thresholded zero-crossings superimposed on the initial velocity data. (d) The smooth optical flow after 1000 iterations. Note that the noise in the lower part of both video images is completely smoothed away. (e) The final piecewise smooth optical flow after 13 analog-digital cycles. The velocity field is subsampled to improve visibility. With the exception of a square appendage at the right hip, the optical flow field shown corresponds to the correct velocity field. The appendage, caused by the edge-detection scheme lumping part of the garbage can in the background with the contour of the person, represents an instance of what psychophysicists term motion capture (Wang, Mathur and Koch, 1989). More recently, we have successfully computed the optical flow field for images with many, partially occluding, moving people.

FIgure 7. Basic resistance element in analog subthreshold CMOS technology. (a) Shows a one-dimensional cut through the resistive network of Figures 2 and 8. The active circuits—termed Hres (Mead, 1989)—within the shaded areas implement a variable nonlinear resistance.

Each transconductance amplifier implements the conductance G (see Figure 8), whose value can be set by V_{conf}. The spatial response of such a network to a point voltage stimulus applied to the left-hand side is shown in (c). In an ideal tap line, the measured voltage values (points) should follow an exponential decay (lines). $V_T - V_{conf}$ sets the decay length L. The current-voltage characteristic of one such resistive element ("Hres") is illustrated in (b). The voltage V_T controls the maximum current and thus the slope of the resistance, which can vary between 100 $k\Omega$ and 10 $G\Omega$. Hres contains $6 + N$ transistors, where N is the number of neighbouring connecting nodes. Many variations are possible.

Figure 8. Circuit design for a resistive network for interpolating and smoothing noisy and sparsely sampled depth measurements. The basic version of this CMOS circuit contains 48 by 48 grid points on a hexagonal lattice. The individual resistive elements with a variable slope controlled by V_T, shown in Figure 7, correspond to the term governing the smoothness, λ.

Figure 9. Partial layout of the CMOS chip implementing the resistive network shown in Figure 8, which interpolates sparsely sampled noisy data. Only seven cells (out of the 48 by 48 cell array) are shown in order to demonstrate the hexagonal grid. Each cell is dominated by the two capacities (approximately $1 - 3$ picofarads each) for holding the depth data and its associated confidence value, consists of 45 transistors, and measures 164 x 125 λ^2. For a $\lambda = 1.0\ \mu m$ production run, the total chip measures 8.5 x 6.5 mm^2. The total chips contains 100,607 transistors. If image acquisition devices (phototransistors) are placed on the chip, the sample- and-hold circuitry can be eliminated, substantially reducing the area of the elementary cell.

Figure 10: Experimental data from the surface interpolation chip of Figure 9. (**a**) As input we used a 1 pixel wide and 48 pixel long line of constant voltage. Voltage along a one-dimensional scan line across the chip is plotted when the input in (**a**) is read in along the X-direction (**b**) or along the Y-direction (**c**). The fact that the plots in (**b**) and (**c**) are not identical reveals consistent

inhomogeneities across the circuit, possibly associated with the scanning circuitry. The voltage response in (c) was averaged 48 times across the chip and is shown in (d) (dots). We fitted the function $V(i) = 2.1 + 0.3e^{-|24-i|}$ through these points (solid line), where i is the node number. The response of a one-dimensional infinite lattice of resistances to a δ function input at location j should be $V(i) = c_1 + c_2 e^{-|i-j|/L}$, where the space constant L is given by the square root of the ratio of the resistance-to-ground to the transversal resistance.

Figure 11: Measured voltage output of the circuit for a zero voltage boundary condition (Dirichlet boundary condition). The input voltage V_{in} is set to the values shown in (a). A 5 pixel wide strip shown in black around the left tower does not contain any data and the conductance G is set to zero at these locations. (b), (c) and (d) shows the measured output for three different settings of the transversal conductance T at the same voltage scale. The final surface f will be flat for large enough values of T (e.g. (d)). The noise is caused primarily by small offset currents which vary from location to location.

Figure 12: Measured voltage output of the circuit for a zero derivative boundary condition (Neumann boundary condition). The input voltage V_{in} is set to the values shown in (a). A 5 pixel wide strip shown in black around the left tower and a 2 pixel wide strip around the entire boundary of the chip does not contain any data. (b), (c) and (d) shows the measured output for three different settings of the transversal conductance T at the same voltage scale. The value of the surface behind the left tower is not pulled to the same value as the tower itself (as it should), due to the leakage effects.

5 Bibliography

A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.* , Vol. 11, 1989, pp. 2–12.

A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, Mass., 1987.

E. Gamble and T. Poggio, "Integration of Intensity Edges with Stereo and Motion," *Artificial Intelligence Lab. Memo No. 970*, MIT, Cambridge, Mass., 1987.

S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Patteern Analysis and Machine Intelligence*, Vol. 6, 1984, pp. 721–741.

W.E.L. Grimson, *From Images to Surfaces*, MIT Press, Cambridge, 1981.

J. Harris, C. Koch, J. Luo and J. Wyatt, "Resistive fuses: analog hardware for detecting discontinuities in early vision. In: *Analog VLSI Implementations of Neural Systems*, Mead, C. and Ismail, M. eds., Kluwer, Norwell, MA, in press.

J. Harris, C. Koch, E. Staats and J. Luo, "Analog hardware for detecting discontinuities in early vision," submitted to *Intl. J. Computer Vision.*

E.C. Hildreth, *The Measurement of Visual Motion*, MIT Press, Cambridge, Mass., 1984.

E.C. Hildreth and C. Koch, "The analysis of visual motion: from computational theory to neuronal mechanism," *Ann. Rev. Neurosci.*, Vol. 10, 1987, pp. 477–533.

B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, 1981, pp. 185–203.

J. Hutchinson, C. Koch, J. Luo and C. Mead, "Computing motion using analog and binary resistive networks," *IEEE Computer*, Vol 21, 1988, pp. 52- 63.

C. Koch, "Seeing chips: Analog VLSI circuits for computer vision," *Neural Computation*, Vol. 1, 1989, pp. 184-200.

C. Koch, J. Marroquin, and A. Yuille, "Analog 'Neuronal' Networks in Early Vision," *Proc. Natl. Academy of Sciences USA*, Vol. 83, 1986, pp. 4263–4267.

D. Marr and E. C. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. Lond. B*, Vol. 207, 1980,

pp. 187–217.

J. Marroquin, "Probabilistic solutions of inverse problems," *A.I. Lab Memo. No. 860*, MIT, Cambridge, 1985.

J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *J. Am. Statistics Assoc.*, Vol. 82, 1987, pp. 76–89.

J.C. Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., Vol. I, 1891, pp. 407–408. Republished by Dover Publications, New York, 1954.

C. A. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley. Reading, Mass., 1989.

H.H. Nagel, "On the Estimation of Optical Flow: Relations Between Different Approaches and Some New Results," *Artificial Intelligence*, Vol. 33, 1987, pp. 299–324.

N.M. Nasrabadi, S.P. Clifford and Y. Liu, "Integration of stereo vision and optical flow by using an energy-minimization approach," *J. Opt. Soc. Am. A*, Vol. 6, 1989, pp. 900-907.

T. Poggio, E.B. Gamble, and J.J. Little, "Parallel integration of vision modules," *Science*, Vol. 242, 1988, pp. 436-440.

T. Poggio, V. Torre, and C. Koch, "Computational Vision and Regularization Theory," *Nature*, Vol. 317, 1985, pp. 314–319.

M.A. Sivilotti, M.A. Mahowald, and C.A. Mead, "Real-Time Visual Computations Using Analog CMOS Processing Arrays," in *1987 Stanford VLSI Conference*, P. Losleben, ed., 1987, pp. 295–312.

S. Ullman, "Analysis of visual motion by biological and computer systems," *IEEE Computer*, Vol. 14, 1981, pp. 57-69.

J.E. Tanner, "Integrated optical motion detection," PhD Thesis, Dept. of Computer Science, Caltech, 1986.

D. Terzopoulos, "Regularization of inverse problems involving discontinuities," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 8, 1986, pp. 413-424.

A. Verri, F. Girosi, and V. Torre, "Mathematical properties of the two- dimensional motion field: from singular points to motion parameters," *J. Opt. Soc. Am. A*, Vol. 6, 1989, pp. 698–712.

A. Verri, A. and T. Poggio, "Motion field and optical flow: qualitative properties," *IEEE Trans.*

Pattern Anal. Machine Intell., Vol 11, 1989, pp. 490-498.

H.T. Wang, B. Mathur and C. Koch, "Computing optical flow in the primate visual system," *Neural Computation*, Vol. 1, 1989, pp. 92- 103.

A.L. Yuille, "Energy Functions for Early Vision and Analog Networks," *Artificial Intelligence Lab. Memo No. 987*, MIT, Cambridge, Mass., 1987.

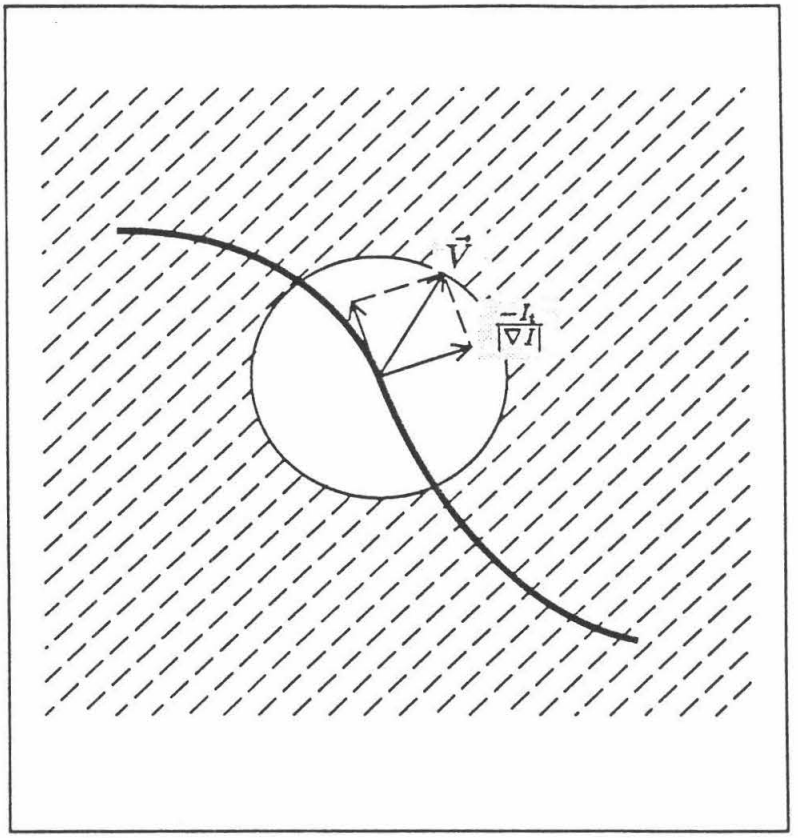


Fig. 1

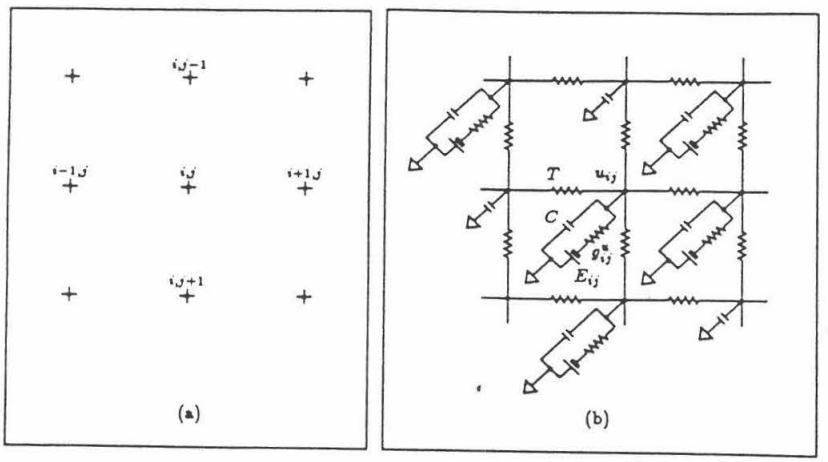
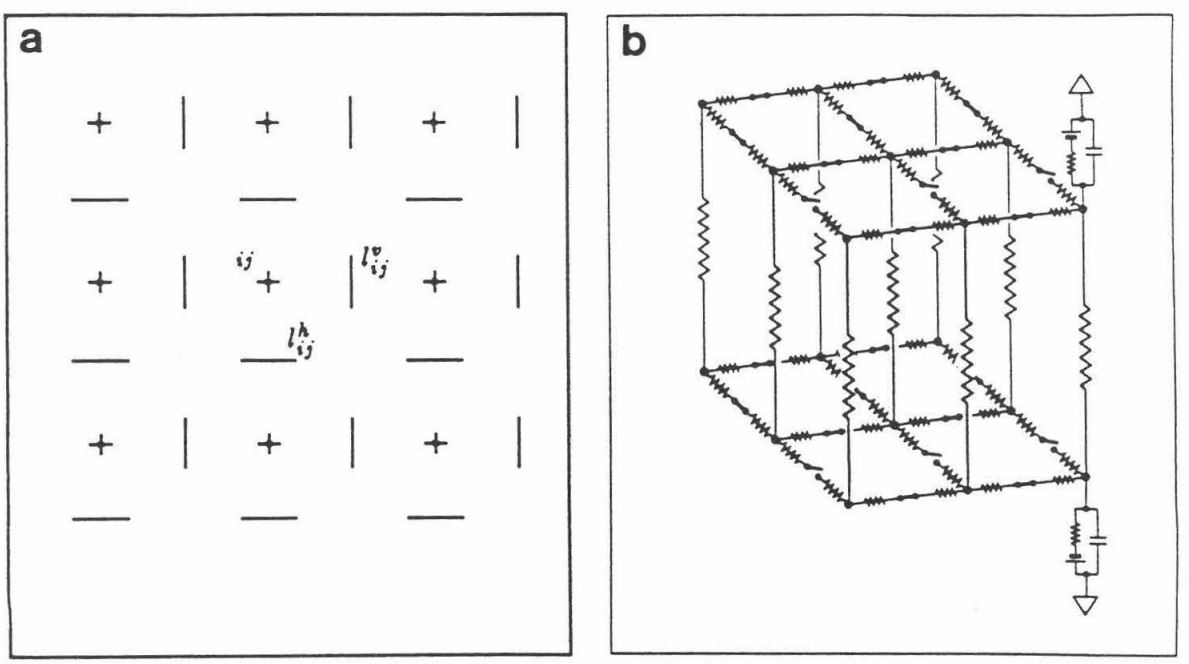
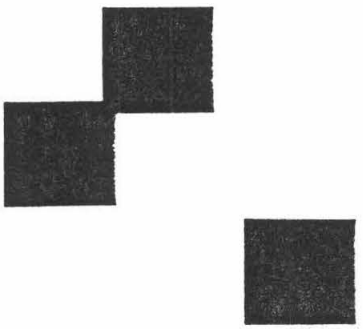


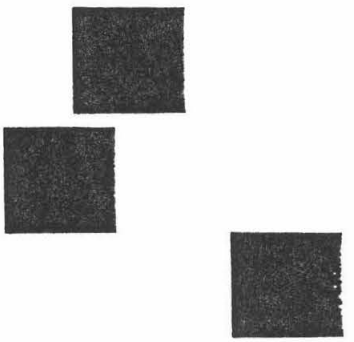
Fig. 2

Fig. 3

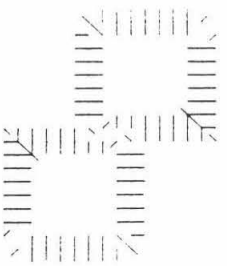




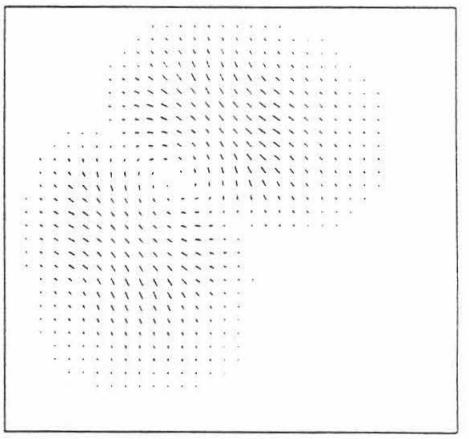
(a)



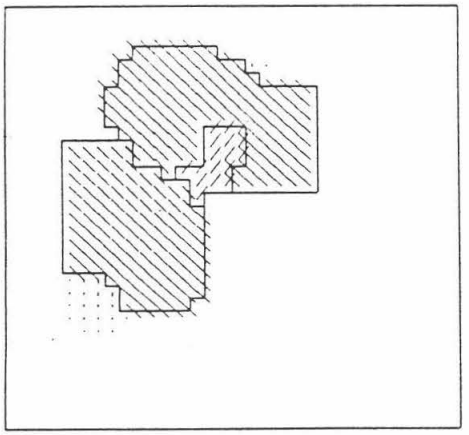
(b)



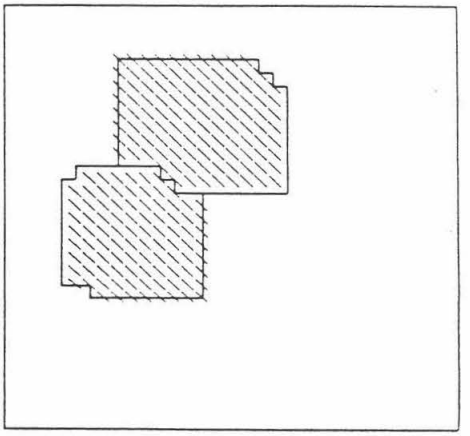
(c)



(d)



(e)



(f)

Fig. 4

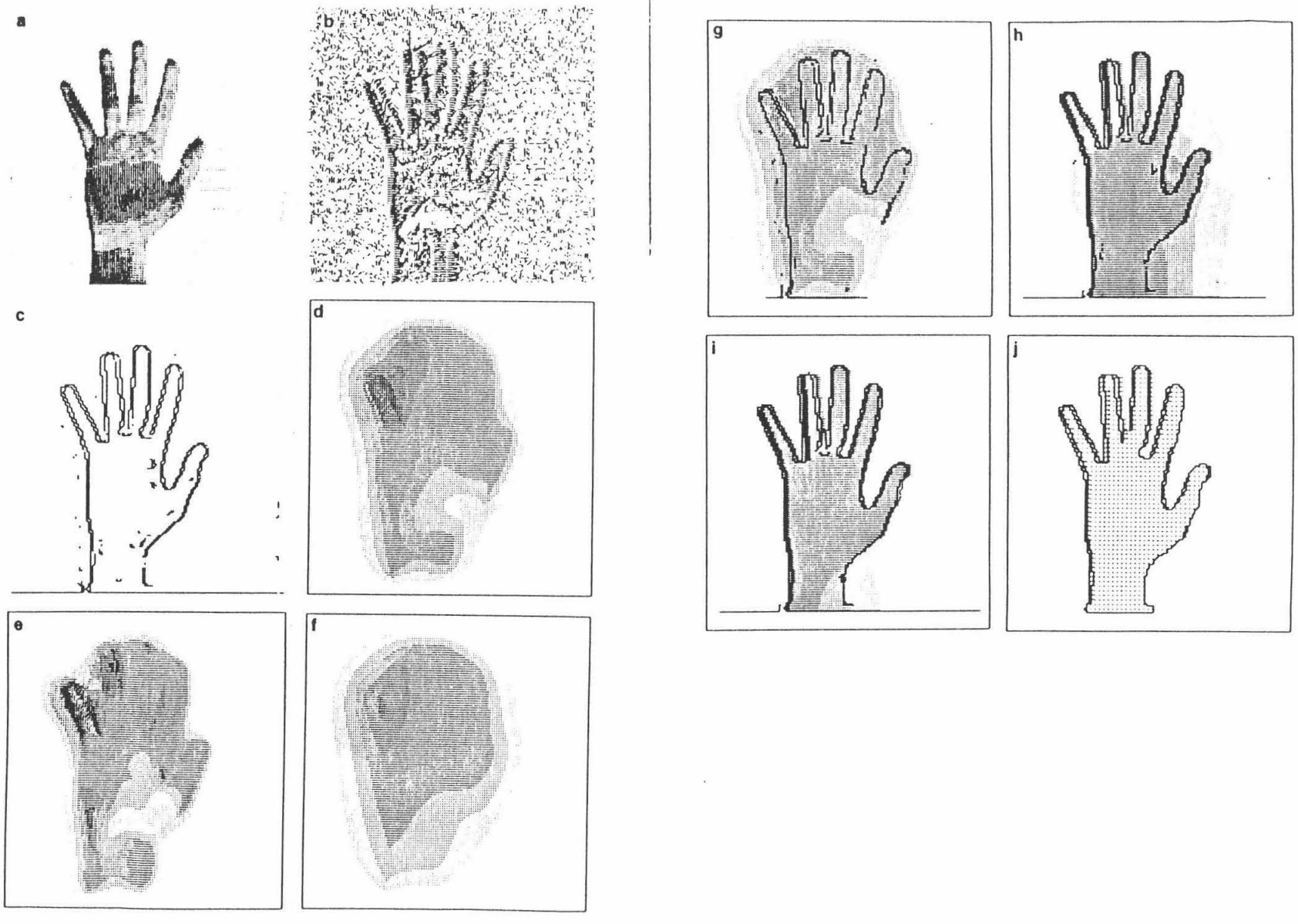
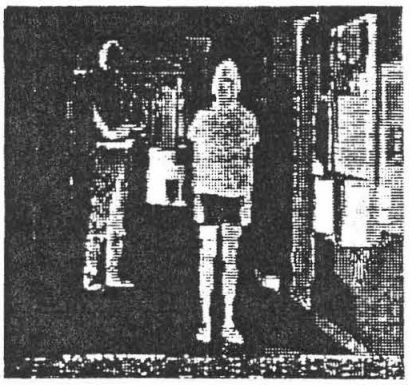
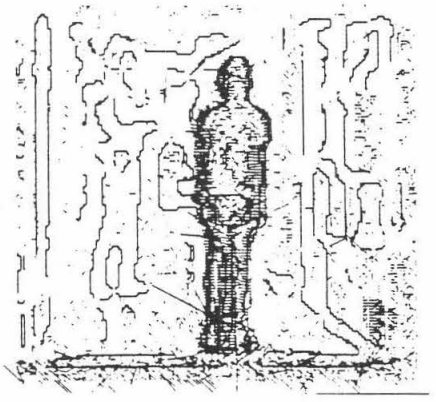


Fig 5

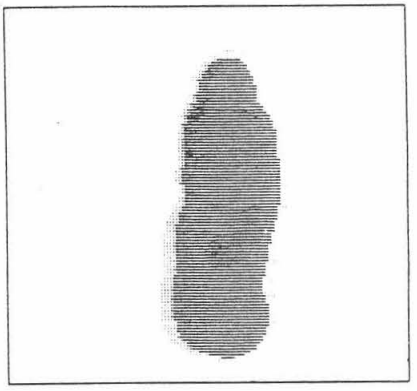


(a)

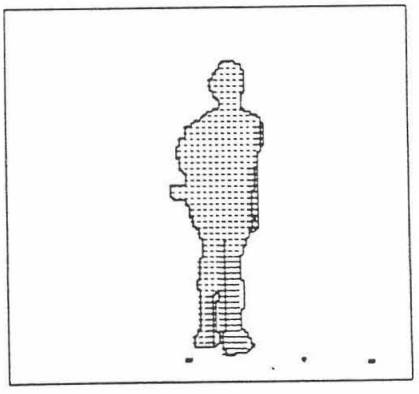
(b)



(c)

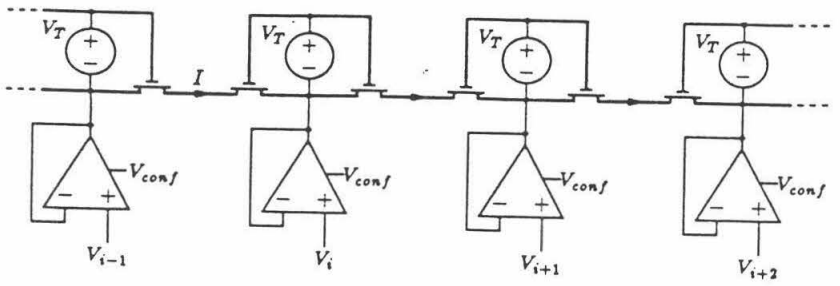


(d)

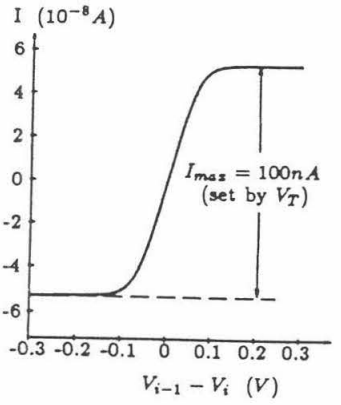


(e)

Fig. 6


V_T
I
V_{conf}
V_{i-1}
V_i
V_{i+1}
V_{i+2}

(a)


I $(10^{-8}A)$
6
4
2
0
-2
-4
-6
$I_{max} = 100nA$
(set by V_T)
-0.3 -0.2 -0.1 0 0.1 0.2 0.3
$V_{i-1} - V_i$ (V)

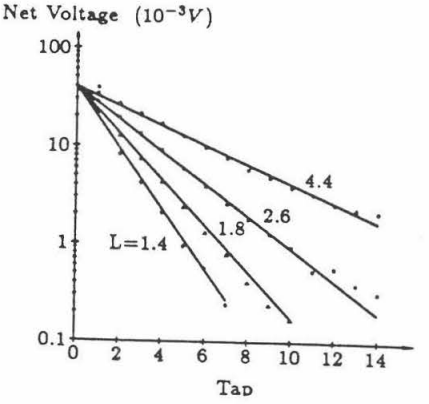

Net Voltage $(10^{-3}V)$
100
10
1
0.1
4.4
2.6
1.8
L=1.4
0 2 4 6 8 10 12 14
Tap

Fig. 7

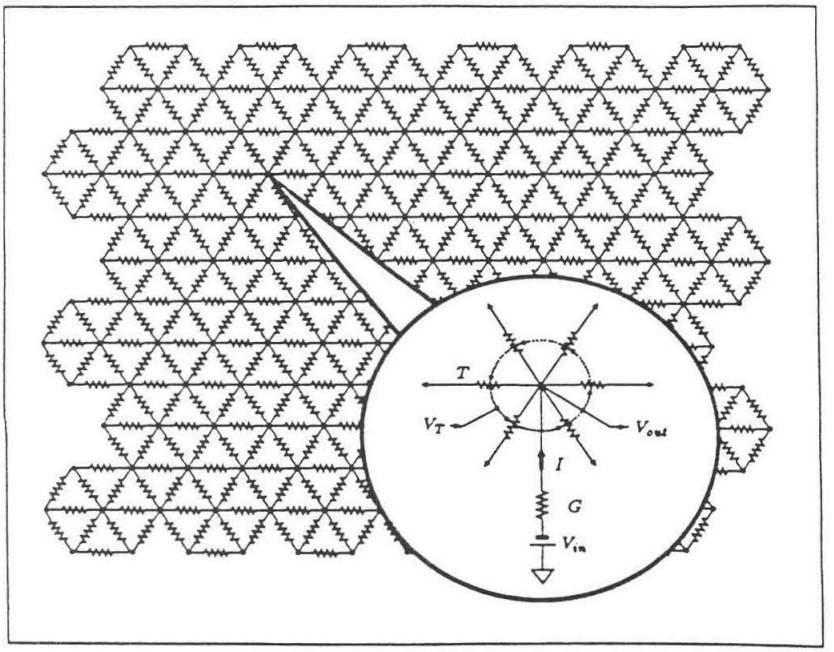

T
V_T
V_{out}
I
G
V_{in}

Fig. 8

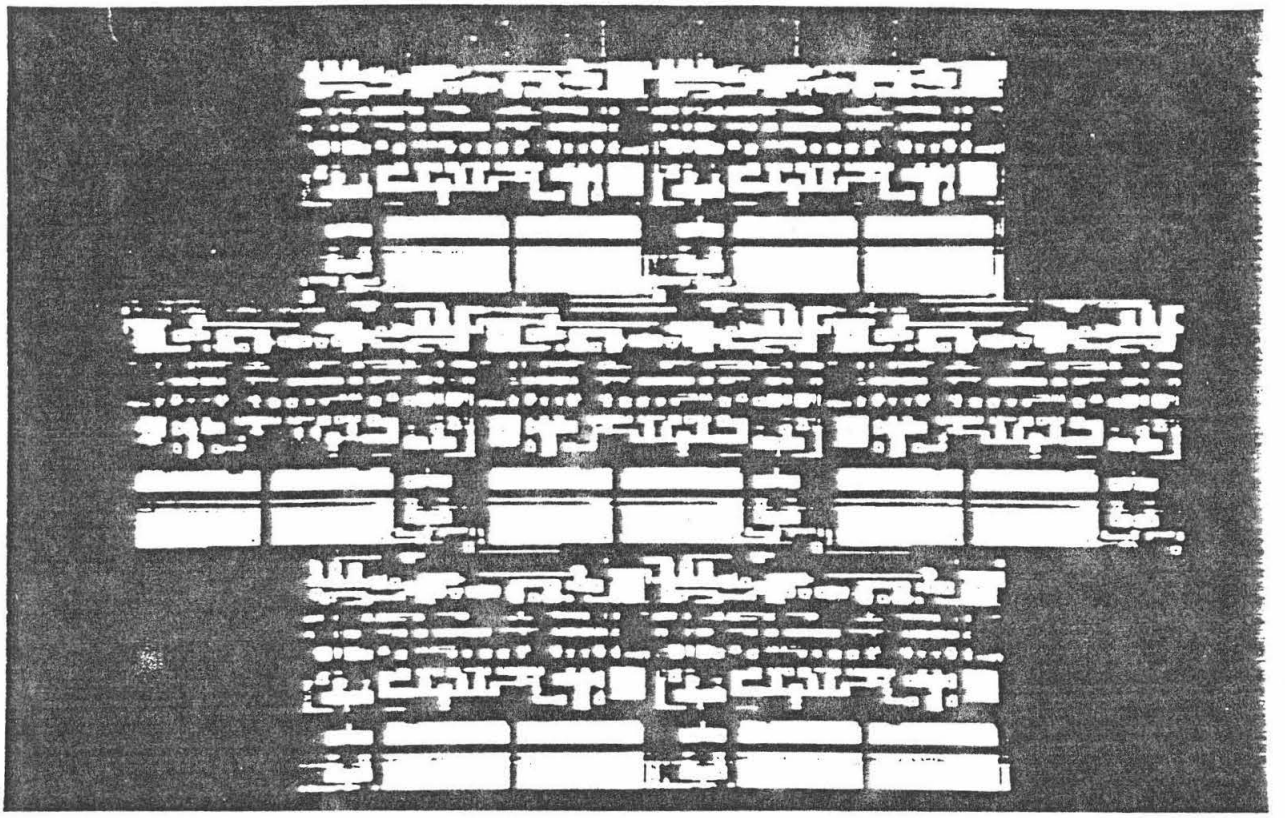


Fig. 9

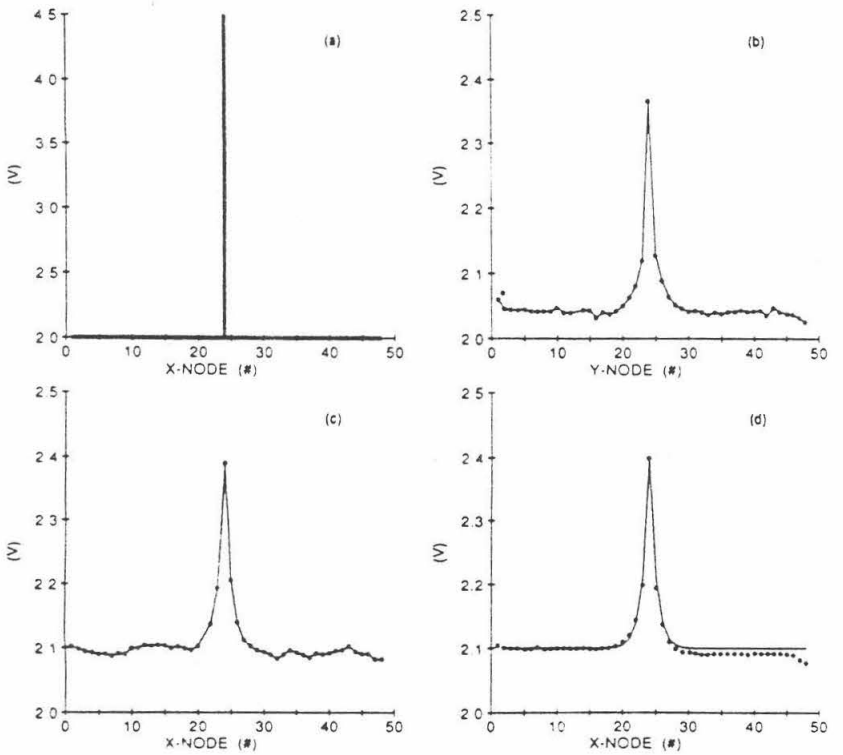


Fig. 10

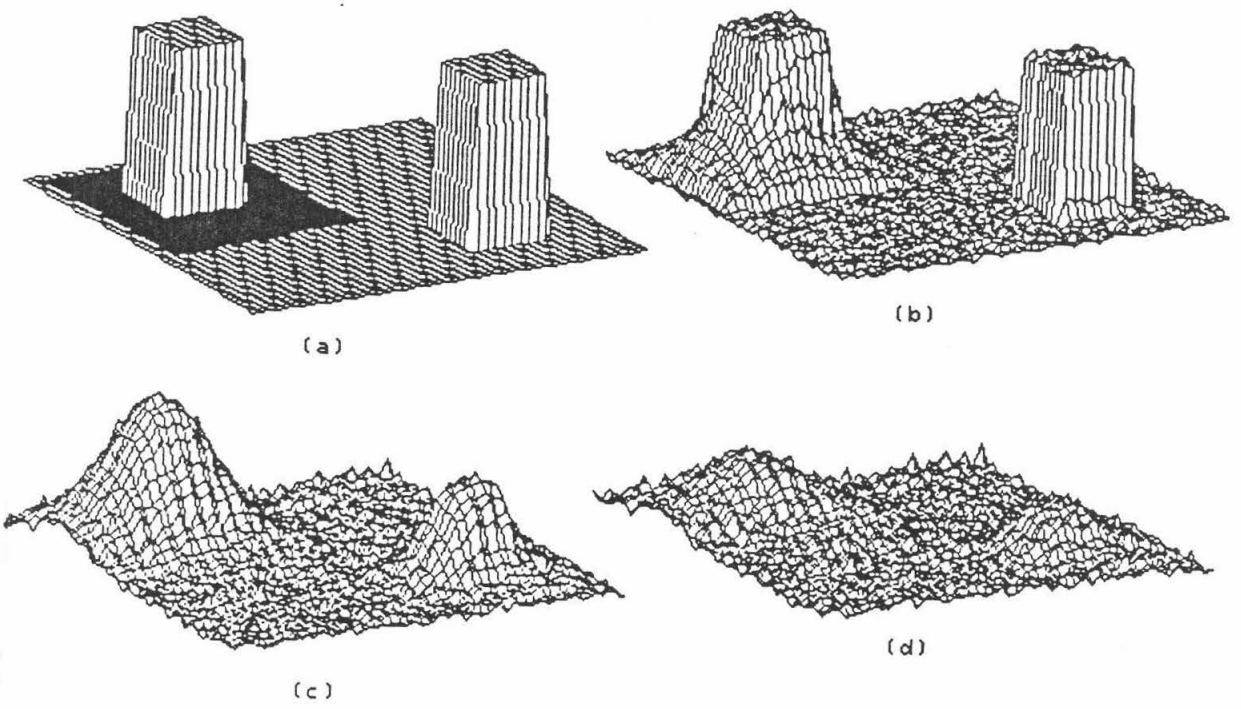


Fig. 11

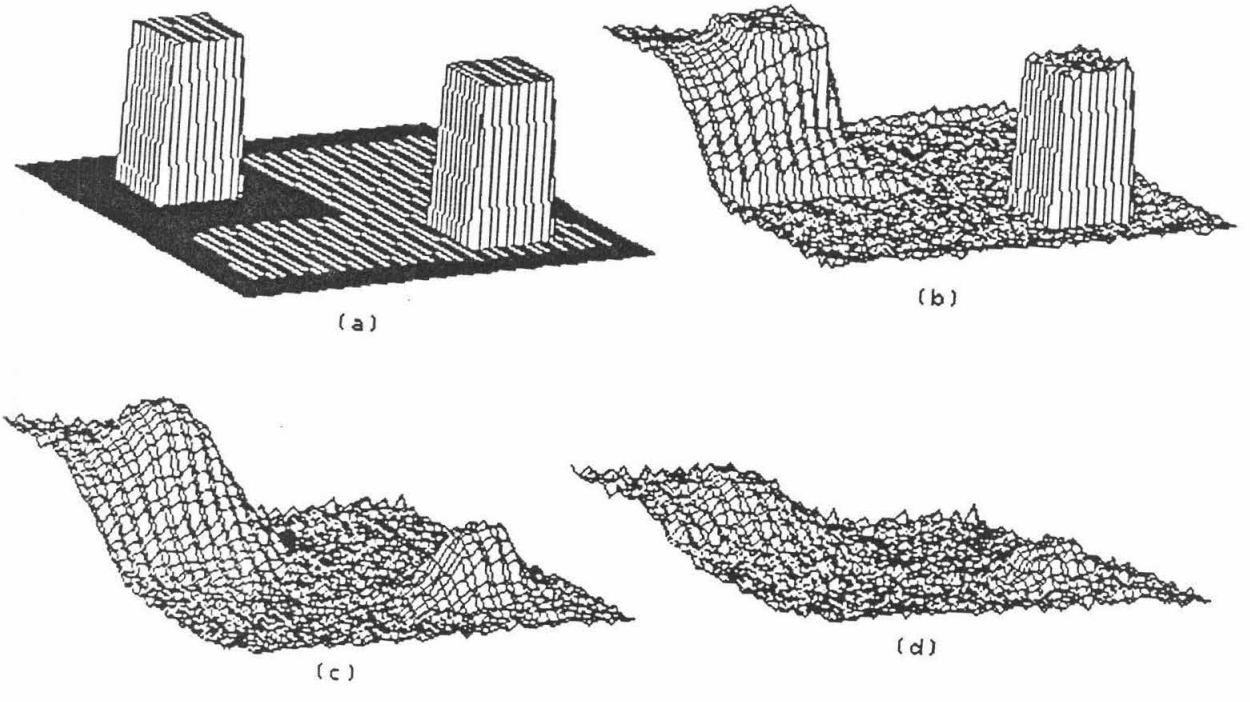


Fig. 12